Neurocomputing 171 (2016) 966–981

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Clustering and pattern search for enhancing particle swarm optimization with Euclidean spatial neighborhood search

Xinchao Zhao [a,*], Wenqiao Lin [b], Junling Hao [c], Xingquan Zuo [d], Jianhua Yuan [a]

[a] School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] State Key Laboratory of Information Photonics & Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[c] School of Statistics, University of International Business and Economics, Beijing 10029, China
[d] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 12 October 2014
Received in revised form
24 May 2015
Accepted 13 July 2015
Communicated by Swagatam Das
Available online 20 July 2015

Keywords:
Particle swarm optimization
Clustering
k-NN
Pattern search
Euclidean spatial neighborhood
Evolutionary algorithm

ABSTRACT

There are many well-known particle swarm optimization (PSO) algorithms which consider ring/star/von Neumann et al. topological neighborhood and scarcely aim at Euclidean spatial neighborhood structure. *k*-Nearest Neighbors (*k*-NN) is a kind of clustering method to find the necessary representatives among a group of objects efficiently. Pattern search (PS) is a successful derivative-free coordinate search method for global optimization. All these observations inspire the innovative ideas to propose an enhanced particle swarm optimization algorithm (*pk*PSO). Particles efficiently explore for the promising areas and solutions with clustering on the Euclidean spatial neighborhood structure. Particle swarm continuously exploits at the just found promising areas with PS strategy at the latter stage of optimization. The cooperative effect of *k*-NN and PS strategies is firstly verified. Based on classical, rotated and shifted benchmarks, extensive experimental comparisons indicate that *pk*PSO has a competitive performance when comparing with the well-known PSO variants and other evolutionary algorithms.
© 2015 Elsevier B.V. All rights reserved.

1. Introduction

For several decades, population-based optimization algorithms have been playing increasingly important roles in many fields [38]. They have always been the active research areas for optimization problems solving since their emergence. Without loss of generality, in this paper, we consider the following numerical optimization problem:

$$\text{Minimize}: \ y = f(\vec{x}), \ \vec{x} \in \Theta \tag{1}$$

where $\Theta \in \mathbb{R}^D$ is a compact set, $\vec{x} = (x_1, x_2, \ldots, x_D)^T$ is decision variable, and D is the dimension of $\vec{x}$, i.e., the number of decision variables. Generally, for each variable, it satisfies a boundary constraint, such that

$$L_j \le x_j \le U_j, \quad j = 1, 2, \ldots, D \tag{2}$$

Particle swarm optimization (PSO) is a swarm intelligence technique developed by Kennedy and Eberhart [22,11], which is a stochastic and population-based adaptive optimization method inspired by social behavior of bird flocks. As one of the versatile and efficient swarm intelligence techniques, PSO has attracted increasing attentions and been widely applied in various areas [38]. The outstanding feature of PSO is its new solution generation mechanism which distinguishes it from other biological-inspired optimization techniques. PSO guides its search direction by this generation strategy in which each particle updates its velocity through a linear combination among its present status, historical best experience and the swarm best experience. Such a velocity updating strategy is easy to achieve, but experimentally inefficient when searching in a complex space. The reason may be that the swarm will converge quickly by tracking only its historical best experience and global best experience. It is easy to fall into local optima due to being lack of an effective escaping mechanism at the latter stage of evolution. Therefore, how to choose the typical and promising representative solutions among the current population and powerful local search techniques to be executed on these promising solutions is essential to the performance of PSO. This is the initial motivation of this research. *k*-NN is adopted to find the promising solutions and pattern search is adopted for exploitation in this paper.

Recently, many improved learning strategies and encouraging PSO variants have been proposed. Liang et al. [25] proposed a novel comprehensive learning strategy for PSO where other particles' previous best positions are exemplars to be learned from by any particle. Each dimension of a particle can potentially learn from a different exemplar. The new strategy makes the particles have more exemplars to learn from and a larger potential space to fly. Wang et al. [39] employed a generalized opposition-based

* Corresponding author. Tel.: +861062281308.
E-mail address: xcbupt@126.com (X. Zhao).

http://dx.doi.org/10.1016/j.neucom.2015.07.025
0925-2312/© 2015 Elsevier B.V. All rights reserved.

learning (GOBL) and Cauchy mutation to provide a faster convergence and help particles escape from local optima. Cho et al. [4] presented a species-based particle swarm optimization to generate new particles for multiple optimal solutions using deterministic sampling. Nickabadi et al. [28] proposed a new adaptive inertia weight by using the success rate of the swarm as its feedback parameter to ascertain the particles' situation in the search space. Zhan et al. [41] put forward an adaptive particle swarm optimization (APSO), in which two main steps are conducted to adaptively adjust the parameters when the swarm lies in a different evolutionary state (exploration, exploitation, convergence and jumping out) in each generation. Then an elitist learning strategy is performed when the evolutionary state is classified as convergence state. Zhan et al. [42] also applied an orthogonal learning strategy to discover more useful information that lies in each particle's historical best experience and its neighborhood's best experience. Zhao et al. [44] proposed a novel multi-swarm cooperative multistage perturbation guiding particle swarm optimizer. Multi-swarm cooperation aims to improve the evolving efficiency via information communicating and sharing among different sub-swarms. Multistage perturbation strategy aims to slow down the learning speed and intensity.

As many excellent contributions indicate, PSO has already been shown to be a promising global and combinatorial optimization algorithm. However, like most population-based algorithms, PSO takes a long time because of its stochastic nature and it is always a challenge to define even more competitive optimization algorithms. To improve the performance of PSO, a new PSO variant (*pk*PSO) is proposed for global optimization. Particle swarm efficiently finds the promising areas and the representatives of solutions at the just found areas with PSO-based framework and k nearest neighbor clustering algorithm. Then each particle tracks its best historical experience from the Euclidean spacial neighbors. It hopes to produce an even better solution with pattern search strategy on the basis of the found-so-far best particles.

The rest of this paper is organized as follows. The framework of canonical PSO and several other state-of-the-art evolutionary algorithms (EAs) are described in Section 2. Then, clustering technique of k-nearest neighbor algorithm and pattern search strategy (PS) are introduced in Section 3. Section 4 introduces the research motivation and algorithmic components of the Euclidean spacial neighborhood-based PSO with clustering and pattern search. In Section 5, comprehensive experimental comparisons are conducted to analyze the key parameters and to verify the efficacy and efficiency of *pk*PSO. Finally, this paper is concluded at Section 6.

2. Population-based evolutionary algorithms

Nature inspired optimization algorithms, such as genetic algorithm (GA) [16], particle swarm optimization algorithm (PSO) [29], differential evolution (DE) [35], ant colony optimization algorithm (ACO) [9] and artificial bee colony algorithm (ABC) [21], have attracted wide attention to researchers from both model and metaphor levels. It inspires us greatly to tackle complex optimization problems. Careful observations on the underlying relations between optimization and biological evolution lead to the prosperous development of evolutionary computation and swarm intelligence.

2.1. Canonical particle swarm optimization

A canonical PSO [29] is an optimization technique based on the cooperation and competition among individuals to search the optimal solution in a D-dimensional hyperspace. There is a swarm of particles and each individual has a fitness value which is decided by the objective function. During the particles' evolution, each particle has a velocity vector $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ and a position vector $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and then flies to the potential optimal position under the guidance of the heuristic information, where i is a positive integer indexing the particle in the swarm.

Particle tracks two extremes to update itself. One is its personal historical best position vector $\vec{p}_i$ and the other is the best position found by the entire swarm, which is denoted as $\vec{p}_g$. The vector $\vec{v}_i$ and the position $\vec{x}_i$ are randomly initialized and updated by the following formulae through the guidance of $\vec{p}_i$ and $\vec{p}_g$:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 (p_{i,d}(t) - x_{i,d}(t)) + c_2 r_2 (p_{g,d}(t) - x_{i,d}(t)) \quad (3)$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad (4)$$

where ω is the inertia weight, coefficients c_1 and c_2 are the cognitive and social weights, and r_1, r_2 are two uniform random numbers within the interval of [0, 1]. The inertia weight ω was introduced by Shi and Eberhart [33,34]. A usually used linearly decreasing strategy of ω is

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \times \frac{t}{T} \quad (5)$$

where t and T are the current and the maximal iteration numbers or the numbers of function evaluations. ω_{max} and ω_{min} are the predefined maximal and minimal inertia constants respectively.

2.2. CLPSO

Comprehensive learning particle swarm optimizer for global optimization of multimodal functions (CLPSO), introduced by Liang et al. [25], is a new variant of PSO aiming at avoiding premature convergence when solving multimodal problems based on a new velocity update equation as follows:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 (p_{f_i(d)}(t) - x_{i,d}(t)) \quad (6)$$

where $\vec{f}_i = [f_i(1), f_i(2), \dots, f_i(D)]$ defines which particle's *pbest* that particle i should follow. $p_{f_i(d)}$ can be the corresponding dimension of any particle's *pbest* including its own *pbest*. The decision depends on probability Pc which is referred to as the learning probability. It takes different values for different particles.

In CLPSO, each dimension of a particle can learn from different *pbest* for different dimensions, instead of learning from its own two exemplars (*pbest* and *gbest*) at the same time. This new learning strategy effectively enhances population diversity and potentially enables swarm diverse to avoid premature convergence.

2.3. CMA-ES

Hansen et al. [15] proposed a new evolutionary algorithm, CMA-ES, which is a second order method. CMA-ES is similar to quasi-Newton method, but not inspired by it. Within an iterative procedure, CMA-ES will estimate the inverse of *Hessian* matrix using the covariance adaptation.

In CMA-ES, any new candidate solutions are sampled according to a multivariate normal distribution $N(\vec{m}_k, \sigma_k^2 \mathbf{C}_k)$ at iteration k. Here, $\vec{m}_k \in \mathbb{R}^D$ is the current best solution, $\sigma_k > 0$ is the step size, and $\mathbf{C}_k$ is a symmetric and positive definite $D \times D$ covariance matrix with $\mathbf{C}_0 = I$. Then, the dependence between any two variables in this distribution is represented by a covariance matrix. Therefore, to update the covariance matrix of this distribution is to promote the information exchange of solutions. So this method is denoted as covariance matrix adaptation (CMA).

Similar to the approximation of the inverse Hessian matrix in the quasi-Newton method, adaptation of the covariance matrix means to learn a second order model of the objective function. But, unlike traditional optimization methods, CMA-ES is a derivative-free approach and fewer assumptions on the objective function are

Download English Version:

https://daneshyari.com/en/article/407470

Download Persian Version:

https://daneshyari.com/article/407470

Daneshyari.com